

AMLEngine-DLL Interfaces

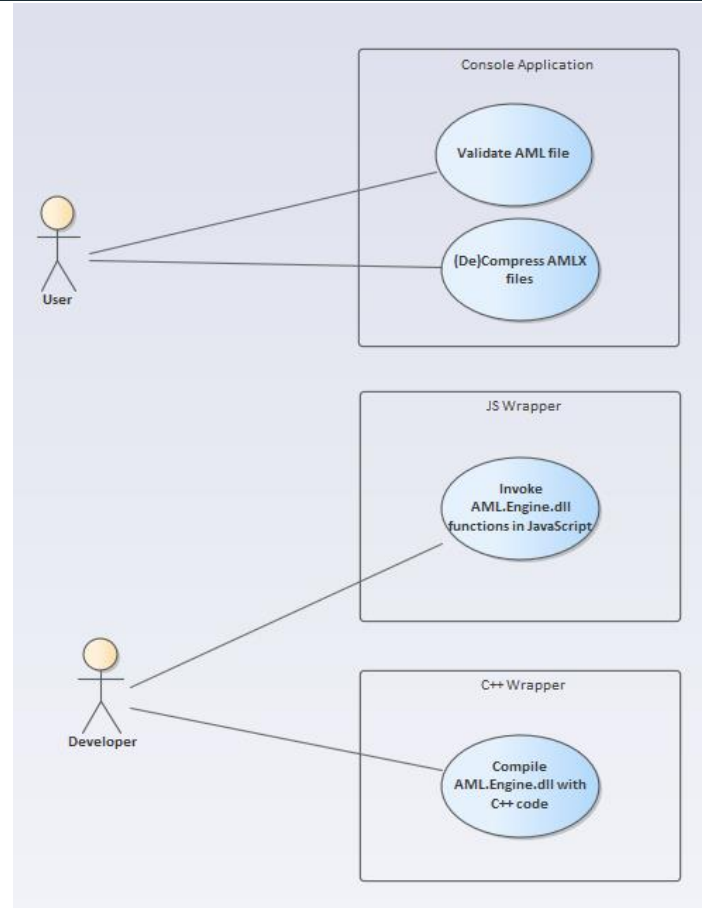
TINF18C (15.05.2020)

Team 4: Joshua, Kevin, Krister, Lucas, Markus, Robin

Vorstellung der Projektmitglieder

					
Joshua	Kevin	Krister	Lucas	Markus	Robin
<i>Test-Manager</i>	<i>Produkt-Manager</i>	<i>Entwickler</i>	<i>Projektleiter</i>	<i>Doku</i>	<i>Leitender Entwickler</i>

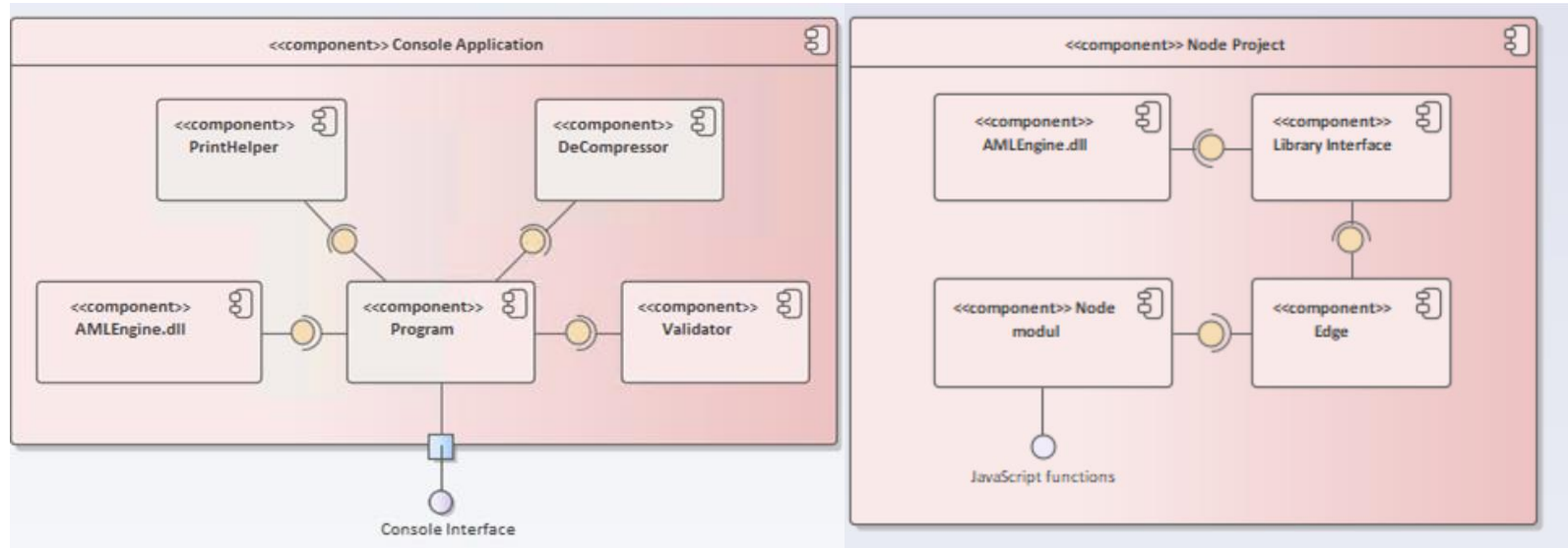
Master Use-case



Produktübersicht

- Implementierung eines JS-Wrappers
 - Standardfunktionen von AML
- Erstellung einer Kommandozeilenapplikation
 - AML(X)-Datei gegen CAEX-Schema validieren
 - Gibt aussagekräftige Fehlermeldungen aus
 - AMLX-Datei entpacken
 - AMLX-Datei in einem Verzeichnis erzeugen
- Erstellen einer Anleitung zur Verwendung der AMLEngine in C++

Systemarchitektur



Modulübersicht

- C++ Wrapper Dokumentation
- JavaScript Wrapper
 - Node Package
 - C# Adapter DLL
- Console Application
 - Program
 - Validator
 - (De-)Compressor

Testing-Vorgehensweise

- Aufteilung der Verantwortlichen
- Erstellung Test-Suites/Test-Cases
- Review
- Durchführen der Test-Cases
- Dokumentation

11.2 Testsuite <TS-002 JS Wrapper>

11.2.1. <TC-002-001> Wrapper import

Term	Value
Testcase ID:	TC-002-001
Testcase Name:	Wrapper import
Req.-ID:	UC.002, LF30, LF40
Description:	Validates that the wrapper package can be downloaded and imported into a node project.

Test Steps

Step	Action	Expected Result
1	Make sure to have all the required software installed. (See 7.3)	n/a
2	Run the command <code>npm i amlenginewrapper --save</code>	Success message is displayed and the package is added to the package.json file
3	Invoke the interactive node terminal using the command <code>node</code>	The console displays the message "Welcome to Node.js"
4	Write the command <code>wrapper=require('amlenginewrapper');</code>	The package is imported without any error messages

5.2 Testsuite <TS-002 JS Wrapper>

5.2.1. <TC-002-001> Wrapper import

Term	Value
Testcase ID:	TC-002-001
Testcase Name:	Wrapper import
Req.-ID:	UC.002, LF30, LF40
Description:	Validates that the wrapper package can be downloaded and imported into a node project.

Test Steps

Step	Action	Expected Result	Actual Result
1	Make sure to have all the required software installed. (See 7.3)	n/a	All the required software was installed.
2	Run the command <code>npm i amlenginewrapper --save</code>	Success message is displayed and the package is added to the package.json file	The success message was displayed and the amlenginewrapper package was added
3	Invoke the interactive node terminal using the command <code>node</code>	The console displays the message "Welcome to Node.js"	The message "Welcome to Node.js" was displayed
4	Write the command <code>wrapper=require('amlenginewrapper');</code>	The package is imported without any error messages	The package was successfully imported

Test result	
Tester:	Joshua
Date:	01.05.2020
Testcase result:	PASS

C++ Wrapper

Project Structure

- C++ Wrapper in the Microsoft World

- i. Preparation

- ii. Install dependencies

- iii. Add sample code

- iv. Compiling the code

Common Compiler Errors

- C++ Wrapper with GNU C++ Compiler

- ◊ Issues and findings when compiling with gcc

- Glossary



Live Demo



Erkenntnisse aus dem Projekt

- “Planung ist das A und O”
 - Meetings min 1/Woche
 - Todos bis zum nächsten Meeting helfen
 - Dokumente sollten reviewed werden
 - Redundanz in Dokumenten erschwert Updates
- Einarbeitung in völlig unbekannte Themen dauert länger als erwartet
- Zeitnahe Kommunikation hilft viel
- GitHub bietet deutlich mehr als reine Versionsverwaltung für Code

Selbstkritik

- Nicht standardisierte Dokumentation
 - z.B. Problematische Commit-Nachrichten -> Ausarbeitung und Einführung von Standards
 - Unterschiedliche Schreibweisen für Fachbegriffe -> Erarbeiten einer gemeinsamen Grundlage
 - Keine festgelegte Sprache
- Workflow teilweise nicht eingehalten
 - z.B. Teilweise Merge ohne Review
 - Kanban Board in unregelmäßig gepflegt

Vielen Dank für Ihre Aufmerksamkeit!

Joshua Franz Testmanager <i>inf18001@lehre.dhbw-stuttgart.de</i>	Kevin Kretschmar Produktmanager <i>kevin.kretschmar@diconium.com</i>	Krister Wolfhard Entwickler <i>inf18164@lehre.dhbw-stuttgart.de</i>
Lucas Krauter Projektleiter <i>inf18223@lehre.dhbw-stuttgart.de</i>	Markus Limbacher Dokumentation <i>markus.limbacher@diconium.com</i>	Robin Beer Leitender Entwickler <i>inf18046@lehre.dhbw-stuttgart.de</i>